#2

Docket No.: 1046.1236/JDH

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
Yutaka HAGA)
) Group Art Unit: Unassigned
Serial No.: To be assigned)
) Examiner: Unassigned
Filed: February 6, 2001)
)
For: APPARATUS FOR COLLECTING) PROFILES OF PROGRAMS)

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-292736
Filed: September 26, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: February 6, 2001

By: _____
James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

CERTIFIED COPY OF
PRIORITY DOCUMENT

OP1137

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ９月２６日

出 願 番 号
Application Number: 特願２０００－２９２７３６

出 願 人
Applicant (s): 富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

２０００年１１月１０日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２０００－

【書類名】　特許願

【整理番号】　0050858

【提出日】　平成12年 9月26日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 11/34

【発明の名称】　プログラム性能情報収集装置及びコンピュータ読取可能な記録媒体

【請求項の数】　5

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　芳賀　豊

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100089244

【弁理士】

【氏名又は名称】　遠山　勉

【選任した代理人】

【識別番号】　100090516

【弁理士】

【氏名又は名称】　松倉　秀実

【連絡先】　０３－３６６９－６５７１

【手数料の表示】

【予納台帳番号】　012092

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　　図面　1

【物件名】　　要約書　1

【包括委任状番号】　9705606

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】プログラム性能情報収集装置及びコンピュータ読取可能な記録媒体

【特許請求の範囲】

【請求項１】プログラムに含まれたサブルーチンの性能情報を収集する装置であって、

前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する、プログラム性能情報収集装置。

【請求項２】サブルーチンの実行体毎に前記性能情報を収集する、請求項１記載のプログラム性能情報収集装置。

【請求項３】特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する、請求項２記載のプログラム性能情報収集装置。

【請求項４】或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する、請求項３記載のプログラム性能情報収集装置。

【請求項５】分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって、

コンピュータに、

前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、

前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、コンピュータシステム上で実行されるプログラムの性能情報(プロフィール)を収集するプログラム性能情報収集装置，及びコンピュータをプログラム性能情報収集装置として機能させるプログラムを記録したコンピュータ読取可能な記録媒体に関する。

【０００２】

【従来の技術】

従来、サブルーチン(「手続」，「関数」等とも呼ばれる)の詳細な性能分析(プロファイリング)を行なう方法として、以下の方法がある。

(１)サブルーチンのプロファイリング用の制御コードを予めプログラム中に埋め込む。

(２)プログラムを解析し、サブルーチンの呼出命令を横取りするようにプログラムを改変する。

【０００３】

【発明が解決しようとする課題】

しかしながら、上記した従来の方法では以下の問題があった。即ち、上記(１)に示したプロファイリング用の制御コードをプログラム中に埋め込む方法では、プログラムの実行時におけるオーバヘッドが大きくなる。このため、(１)の方法は、開発段階でのプログラムに適用され、埋め込まれた制御コードは、出荷される最終製品のプログラムから省かれることが多かった。従って、(１)の方法を最終製品のプログラムに適用することは困難であった。

【０００４】

また、上記(２)に示した呼出命令を横取りするようにする方法では、プログラムコードを改変する。このため、プログラム自身の正当性検査で改変が検出された場合に、検出された改変が不当と判定され、プログラムが動作しなくなる可能性があった。

【０００５】

本発明の目的は、プログラムに改変を加えることなくサブルーチンの性能情報を収集することができるプログラム性能情報収集装置及びコンピュータ読取可能な記録媒体を提供することである。

【０００６】

【課題を解決するための手段】

本発明は、上述した目的を達成するために以下の構成を採用する。即ち、本発明は、プログラムに含まれたサブルーチンの性能情報を収集する装置であって、前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する。

【０００７】

本発明によれば、割込によって性能情報(プロフィール)を収集するので、プログラムに制御コードを埋め込んだり、改変を加えたりする必要がない。従って、例えば、最終製品のプログラム中のサブルーチンの性能情報を収集し、分析することが可能となる。性能情報は、例えば、サブルーチンの累計実行時間，呼出回数，オーバヘッド等である。

【０００８】

本発明は、サブルーチンの実行体毎に前記性能情報を収集する構成としても良い。このようにすれば、詳細なサブルーチンの性能情報を収集することができ、詳細なプログラムの分析が可能となる。実行体は、プロセス(「タスク」とも呼ばれる)或いはスレッドである。

【０００９】

本発明は、さらに、特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する構成としても良い。このようにすれば、さらに詳細なサブルーチンの性能情報を収集することができ、詳細なプログラムの分析を行うことが可能となる。

【００１０】

本発明は、さらに、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する構成としても良い。このようにすれば、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された(ネスティングによって呼び出された)場合の性能情報とを、個別に収集することができ、詳細なプログラムの分析が可能となる。

【００１１】

本発明は、さらに、他のサブルーチンから呼び出された場合の性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて保持する構成としても良い。このようにすれば、各サブルーチン間の呼出関係(ネスティングの状態)を含む詳細な性能情報を得ることができ、詳細なプログラムの分析が可能となる。

【００１２】

また、本発明は、分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって、コンピュータに、前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可能な記録媒体である。

【００１３】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

【００１４】

〔第１実施形態〕

〈ハードウェア構成〉

図１は、本発明の実施形態によるプログラム性能情報収集装置として機能するコンピュータ１のハードウェアブロック図である。コンピュータ１は、例えば、パーソナルコンピュータ(ＰＣ)，ワークステーション(ＷＳ)を用いて構成される。コンピュータ１は、ＣＰＵ２，メインメモリ(MM)３，二次記憶装置４，キーボード５，ポインティングデバイス６及びディスプレイ７が相互に接続されてなる。

【００１５】

二次記憶装置４は、本発明によるコンピュータ読取可能な記録媒体に相当し、

半導体メモリ(例えば、ＲＯＭ，ＲＡＭ，ＳＲＡＭ，フラッシュメモリ，ＥＰＲＯＭ，ＥＥＰＲＯＭ)，磁気ディスク(例えば、ハードディスク，フロッピーディスク)，光ディスク(例えば、ＣＤ-ＲＯＭ，ＰＤ)，光磁気ディスク(ＭＯ)等の各種の記録媒体を用いて構成される。二次記憶装置４は、オペレーティングシステム(ＯＳ)８と、少なくとも１つのアプリケーションプログラム９(以下、「アプリケーション９」という)と、性能分析用のプログラム１０(以下、「分析プログラム１０」という)がインストールされている。また、二次記憶装置４は、各分析プログラム８，９，１０の実行時に使用されるデータを保持している。

【００１６】

ＯＳ８は、コンピュータ１のハードウェア(ＣＰＵ２,ＭＭ３等)とアプリケーション９との間に介在し、ハードウェアの違いを吸収する。また、ハードウェアの実行制御や管理を行う。ＯＳ８は、アプリケーション９を実行するための少なくとも１つの実行体１９を生成する(図２参照)。図２には、例として、３つの実行体１９(１９Ａ，１９Ｂ，１９Ｃ)が示されている。

【００１７】

アプリケーション９は、ＯＳ８上で実行される応用プログラムであり、本発明における分析対象のプログラムに相当する。アプリケーション９は、メインルーチン及び単数又は複数のサブルーチンとの組み合わせ，或いは、複数のサブルーチンの組み合わせで構成される。この例では、アプリケーション９は、メインルーチンと複数のサブルーチンとの組み合わせで構成されている。

【００１８】

分析プログラム１０は、アプリケーション９のサブルーチンに関する性能情報(プロフィール)を収集するためのプログラム，即ち、コンピュータ１を本発明のプログラム性能情報収集装置として機能させるためのプログラムである。

【００１９】

分析プログラム１０は、ＣＤ-ＲＯＭやフロッピーディスク等の可搬性を有する記録媒体から二次記憶装置４(例えば、ハードディスク)にインストールするようにしても良く、コンピュータ１が分析プログラム１０を通信回線を通じて他のコンピュータからダウンロードし、二次記憶装置４にインストールするようにし

ても良い。

【００２０】

キーボード５及びポインティングデバイス６は、コンピュータ１にコマンドやデータを入力するために使用される。以下、キーボード５とポインティングデバイス６とをまとめて指す場合には、「入力装置１１」と表記する。ポインティングデバイス６は、例えば、マウス，ジョイスティック，トラックボール，フラットスペースである。

【００２１】

ディスプレイ７は、陰極線管(ＣＲＴ)，液晶ディスプレイ(ＬＣＤ)，プラズマディスプレイ等を用いて構成されており、ユーザがデータやコマンドを入力するための情報や、プログラムの実行結果等を表示する。

【００２２】

ＭＭ３は、ＣＰＵ２の作業領域として使用される。また、ＭＭ３は、ディスプレイ７に情報を表示するためのデータを保持するビデオメモリ(ＶＲＡＭ)としても機能する。

【００２３】

ＣＰＵ２は、二次記憶装置４に保持されたＯＳ８，アプリケーション９，分析プログラム１０をＭＭ３にロードして実行する。ＣＰＵ２がＯＳ８上でアプリケーション９を実行しているときに分析プログラム１０を実行することにより、コンピュータ１は、本発明のプログラム性能情報収集装置として機能する。

【００２４】

〈機能的構成〉

図２は、図１に示したコンピュータ１によって実行されるプログラム性能情報収集装置の機能ブロック図である。図２(Ａ),(Ｂ)に示すように、プログラム１０がＣＰＵ２によって実行されると、実行環境設定部１５と、命令分析部１６と、性能情報収集部１７とが実現され、少なくとも１つの制御表１８が作成される。以下、各部１５，１６，１７の機能を説明する。

【００２５】

(実行環境設定部)

図２(Ａ)に示すように、プログラム１０がＣＰＵ２によって実行されると、実行環境設定部１５が起動する。実行環境設定部１５は、入力装置１１を通じたユーザからの操作に従って、アプリケーション９の性能情報を収集するための環境設定をＣＰＵ２に施す。

【００２６】

現在コンピュータ１に適用されるあらゆるＣＰＵ２は、プログラムの実行に際して割込を発生させる機能を有している。ＰＣ等で良く使用されているＸ８６アーキテクチャを有するＣＰＵは、１つの命令の実行時にシングルステップ割込を発生させることができる。また、分岐命令が実行された場合にのみ割込を発生させる機能をさらに有するＣＰＵもある。

【００２７】

上記機能を有するＣＰＵとして、例えば、Intel社の“Pentium Pro”を挙げることができる。また、“Pentium Pro”以降の“Pentium II”や“Pentium III”も同様の機能を有している。

【００２８】

“Pentium Pro”は、EFLAGSレジスタと呼ばれる各種のシステム・フラグを含むレジスタを有している。このEFLAGSレジスタの８番目のビット(ビット８)は、“ＴＦフラグ”と呼ばれている。ＴＦフラグがセットされると、デバッグのためのシングル・ステップ・モードがイネーブルとなる。逆に、ＴＦフラグがクリアされると、デバッグのためのシングル・ステップ・モードがディスエーブルとなる。

【００２９】

シングル・ステップ・モードでは、プロセッサ(ＣＰＵ)は、各命令の後にデバッグ例外を生成する。このため、各命令後にプログラムの実行状態を調べることができる。アプリケーションプログラムがＰＯＰＦ，ＰＯＰＦＤ，またはＩＲＥＴ命令を使用してＴＦフラグをセットすると、ＰＯＰＦ，ＰＯＰＦＤ，またはＩＲＥＴ命令の後に続く命令後に、デバッグ例外が発生する。

【００３０】

また、“Pentium Pro”は、ＴＦフラグを用いたデバッグ機能よりも詳細なデ

バッグ機能を設定するレジスタとして、DebugCtlMSRレジスタを有している。DebugCtlMSRレジスタは、最新の分岐，割込，及び例外の記録をイネーブルにする。また、実行された分岐に対するブレイクポイント，ブレイクポイント報告ピン，及びトレースメッセージをイネーブルにする。

【００３１】

DebugCtlMSRレジスタには、特権レベル０での動作時又は実アドレスモード時にＷＲＭＳＲ命令を使用して書くことができる。DebugCtlMSRレジスタへのユーザアクセスを可能にするには、保護モードのオペレーティング・システム・プロシージャを要する。

【００３２】

DebugCtlMSRレジスタは、ＬＢＲ(最新分岐／割込／例外)フラグ(ビット０)と、ＢＴＦ(分岐シングル・ステップ実行)フラグ(ビット１)とを含んでおり、本発明はこれらのフラグを利用する。ＬＢＲフラグがセットされると、プロセッサ(ＣＰＵ)はデバッグ例外が発生する前に最後に行った分岐，及び例外又は割込処理のソースアドレスとターゲットアドレスとを記録する。

【００３３】

一方、ＢＴＦフラグがセットされると、プロセッサ(ＣＰＵ)は、ＥＦＬＡＧＳレジスタのＴＦフラグを“命令シングル・ステップ実行”フラグとしてではなく、“分岐シングル・ステップ実行”フラグとして取り扱う。ＬＢＲフラグ及びＢＴＦフラグの適宜の利用により、分岐処理をＣＰＵにシングル・ステップ実行させることが可能となる。分岐に対するデバッグ・ブレイクポイントをイネーブルするには、実行環境設定部１５は、ＢＴＦフラグ及びＴＦフラグの双方をセットしなければならない。

【００３４】

実行環境設定部１５は、上述したＬＢＲフラグ，ＢＴＦフラグ及びＴＦフラグを組み合わせて設定することにより、分岐命令実行時にのみ割込を発生させ、その時の分岐元アドレス及び分岐先アドレスを取得できるようにする。

【００３５】

即ち、実行環境設定部１５は、被分析対象のプログラム(アプリケーション９)

がＯＳ８上で実行するための実行体１９が生成されるときに、その実行体１９の実行コンテキストを設定するとともに、ＣＰＵ２の分岐命令実行時における割込発生機能を有効化する。

【００３６】

なお、実行環境設定部１５は、自動的に上記環境設定を実行するように構成されていても良い。また、実行環境設定部１５以外の方法で上記設定をＣＰＵ２に施すことができる場合には、分析プログラム１０によって実行環境設定部１５が実現されるようにしなくても良い。

【００３７】

（命令分析部）

図２（Ｂ）に示すように、アプリケーション９の実行時に分析プログラム１０が実行されると、命令分析部１６と、性能情報収集部１７とが実現される。命令分析部１６は、割込が発生した場合にＣＰＵ２から制御を受け取るように、ＣＰＵ２の割込ハンドラとして構成される。

【００３８】

命令分析部１６は、割込の発生によってＣＰＵ２から制御を受け取る場合に、実行されたアプリケーション９中の分岐命令の分岐元アドレス及び分岐先アドレスをＣＰＵ２から受け取る。

【００３９】

命令分析部１６は、分岐元アドレス及び分岐先アドレスを取得すると、分岐元アドレスの命令コードをＭＭ３から読み出し、読み出した命令コードをデコードすることによって分岐命令の種別を判別する。分岐命令の種別は、サブルーチンの呼出命令，復帰命令，ジャンプ命令からなる。

【００４０】

命令分析部１６は、命令種別が呼出命令又は復帰命令である場合には、分岐元アドレス，分岐先アドレス及び分岐命令種別（呼出命令又は復帰命令）を性能情報収集部１７に与える。

【００４１】

なお、ＣＰＵ２によっては、一旦割込が発生すると、実行コンテキストから再

び割込が発生しないように、制御レジスタの設定をクリアする場合がある。この場合、継続して分岐命令実行時に割込みが上がるように、対象プログラムへの復帰時の実行コンテキストを再設定する必要がある。

【００４２】

（性能情報収集部）

性能情報収集部１７は、命令分析部１６から得た分岐元アドレス，分岐先アドレス及び分岐命令種別に加え、ＯＳ８上の実行体識別情報（実行体ＩＤ）を取得する。即ち、性能情報収集部１７は、割込発生時に、ＯＳ８によって提供されるシステムインターフェイスにアクセスすることによって、実行体ＩＤ（プロセスＩＤ，スレッドＩＤ又はタスクＩＤ）を取得する。或いは、性能情報収集部１７は、割込発生時に、ＯＳ８の制御情報に直接アクセスして実行体ＩＤを取得するようにしても良い。

【００４３】

続いて、性能情報収集部１７は、取得した実行体ＩＤを用いてサブルーチンの実行体１９を識別する。これによって、実行体１９毎に性能情報を収集することが可能となる。

【００４４】

なお、この例では、性能情報収集部１７は、アプリケーション９の実行時に生成される全ての実行体１９について性能情報を収集する。これに代えて、少なくとも１つの特定の実行体１９について性能情報を収集するようにしても良い。

【００４５】

性能情報収集部１７は、実行体１９を特定すると、特定した実行体１９によって実行されるサブルーチンの性能情報を記憶する性能情報記憶部としての制御表１８を作成又は更新する。

【００４６】

（制御表）

図３は、性能情報収集部１７によって作成・更新される各制御表１８の構成例を示す図である。制御表１８は、ＭＭ３又は二次記憶装置４上に作成される。図３において、制御表１８は、１つの実行体管理表（execution managing table: E

MT)２１と、少なくとも１つのサブルーチン管理表(subroutine managing table: SMT)２２と、必要に応じて作成される少なくとも１つの呼出管理表(call managing table: CMT)２３とからなる。

【００４７】

ＥＭＴ２１は、特定された実行体１９によって実行されるサブルーチンについてのＳＭＴ２２及びＣＭＴ２３を管理するためのテーブルである。ＥＭＴ２１は、“実行体ＩＤ”と、“次ＥＭＴポインタ”と、“現ＳＭＴポインタ”と、“先頭ＳＭＴポインタ”とを保持する。

【００４８】

“実行体ＩＤ”は、実行体１９の識別情報である。“次ＥＭＴポインタ”は、ＥＭＴ２１に隣接する他の制御表１８中のＥＭＴ２１のアドレスを示す。“現ＳＭＴポインタ”は、現在の性能情報収集部１７のアクセス対象に相当するＳＭＴ２２(現在呼び出されているサブルーチンに対応するＳＭＴ２２)のアドレスを示す。“先頭ＳＭＴポインタ”は、複数のＳＭＴ２２がある場合に、これらの先頭に位置するＳＭＴ２２のアドレスを示す。

【００４９】

ＳＭＴ２２は、各サブルーチンの性能情報を管理するためのテーブルであり、実行体１９によって実行されるサブルーチン毎に作成される。即ち、ＳＭＴ２２は、１つのＥＭＴ２１に対して、そのＥＭＴ２１によって管理される実行体１９によって実行されるサブルーチンの数だけ作成される。

【００５０】

ＳＭＴ２２が複数作成された場合には、ＳＭＴ２２間は相互リンクで接続され、ＳＭＴ２２間を行ったり来たりすることができる。また、あるＳＭＴ２２から他のＳＭＴ２２を辿って目的のＳＭＴ２２へ到達することができる。ＳＭＴ２２間の双方向リンクはサブルーチンの呼出しが発生した場合に相互に連結され、復帰時に解消される。

【００５１】

各ＳＭＴ２２は、“サブルーチンアドレス”と、“ＳＭＴ双方向リンクポインタ”と、“先頭ＣＭＴポインタ”と、“現ＣＭＴポインタ”と、“呼出回数”と

、“累計実行時間”と、“最後の呼出時刻”とを保持する。“呼出回数”，“累計実行時間”及び“最後の呼出時刻”がサブルーチンの性能情報(プロフィール)に相当する。

【００５２】

“サブルーチンアドレス”は、ＥＭＴ２１の実行体ＩＤで特定される実行体１９によって実行されるサブルーチンのアドレスを示す。“ＳＭＴ双方向リンクポインタ”は、複数のＳＭＴ２２を相互リンクで接続した場合に設定されるポインタであり、ＳＭＴ２２自身のアドレスと、このＳＭＴ２２に相互リンクが設定されている少なくとも１つの他のＳＭＴ２２のアドレスとを示す。

【００５３】

“先頭ＣＭＴポインタ”は、このＳＭＴ２２の下位に存する少なくとも１つのＣＭＴ２３のうち、先頭に位置するＣＭＴ２３のアドレスを示す。“現ＣＭＴポインタ”は、性能情報収集部１７の現在のアクセス対象に相当するＣＭＴ２３のアドレスを示す。

【００５４】

“呼出回数”は、ＥＭＴ２１の実行体ＩＤで特定される実行体１９からの当該サブルーチンの呼出回数を示す。“累計実行時間”は、当該実行体１９による当該サブルーチンの実行時間の合計値を示す。“最後の呼出時刻”は、当該実行体１９によって当該サブルーチンが最後に呼び出された時刻である。

【００５５】

ＣＭＴ２３は、あるサブルーチンから他のサブルーチンが呼び出された(ネストした)場合に、呼出元のサブルーチンと呼出先のサブルーチンとの関係(呼出関係)と、呼出先のサブルーチンに対する性能情報を管理するために作成されるテーブルである。

【００５６】

ＣＭＴ２３は、１つのＳＭＴ２２で管理されるサブルーチンによって呼び出される他のサブルーチンの数を最低数として作成される。現在或るサブルーチンから他のサブルーチンが呼び出されている場合、対応するＣＭＴ２３は、呼出元のサブルーチンに対応するＳＭＴ２２中の“現ＣＭＴポインタ”で示される。

【００５７】

各ＣＭＴ２３は、“分岐元アドレス”と、“分岐先アドレス”と、“呼出先ＳＭＴポインタ”と、“呼出回数”と、“累計実行時間”と、“最後の呼出時刻”と、“次ＣＭＴポインタ”とを保持する。“呼出回数”，“累計実行時間”及び“最後の呼出時刻”がサブルーチンの性能情報(プロフィール)に相当する。

【００５８】

“分岐元アドレス”は、呼出元に相当するサブルーチンのアドレスを示す。“分岐先アドレス”は、呼出先に相当するサブルーチンのアドレスを示す。“呼出先ＳＭＴポインタ”は、呼出先のサブルーチンに対応するＳＭＴ２２のアドレスを示す。これらによって、サブルーチン間の呼出関係が保持・管理される。

【００５９】

“呼出回数”は、呼出元のサブルーチンから呼出先のサブルーチンが呼び出された回数を示す。“累計実行時間”は、呼出先のサブルーチンの実行時間の合計値を示す。“最後の呼出時刻”は、呼出元のサブルーチンから呼出先のサブルーチンを最後に呼び出した時刻を示す。“次のＣＭＴポインタ”は、このＣＭＴ２３の次に配置された(隣接する)他のＣＭＴ２３のアドレスを示す。

【００６０】

図３に示す例では、或る実行体１９(例えば、図２に示す実行体１９Ａ)についてのＥＭＴ２１(ＥＭＴ２１Ａ)が作成され、ＥＭＴ２１Ａで管理される実行体１９Ａによって実行される各サブルーチンの性能情報を管理するＳＭＴ２２として、各ＳＭＴ２２Ａ，２２Ｂ，２２Ｃが作成されている。

【００６１】

さらに、ＳＭＴ２２Ａで管理されるサブルーチンに呼び出される他のサブルーチンの性能情報を管理するためのＣＭＴ２３Ａと、ＳＭＴ２２Ｂで管理されるサブルーチンによって呼び出される他のサブルーチンの性能情報を管理するためのＣＭＴ２３Ｂが作成されている。

【００６２】

ＥＭＴ２１Ａは、“次ＥＭＴポインタ”でアクセス可能な他の実行体１９を管理するＥＭＴ２１Ｂに接続されており、性能情報収集部１７は、ＥＭＴ２１Ａを

通じて次のＥＭＴ２１Ｂにアクセスすることができる。

【００６３】

ＳＭＴ２２Ａは、ＥＭＴ２１Ａに対する先頭ＳＭＴに相当する。ＳＭＴ２２Ａは、ＳＭＴ２Ａの次のＳＭＴに相当するＳＭＴ２２Ｂに相互リンクで接続され、ＳＭＴ２２Ｂは、ＳＭＴ２２Ｂの次のＳＭＴに相当するＳＭＴ２２Ｃに相互リンクで接続される。

【００６４】

性能情報収集部１７は、所望のＳＭＴ２２にアクセスする場合には、ＥＭＴ２１Ａの“先頭ＳＭＴポインタ”を用いてＳＭＴ２２Ａにアクセスし、続いて、ＳＭＴ２２Ａ，２２Ｂ，２２Ｃの“ＳＭＴ双方向リンクポインタ”を用いて所望のＳＭＴのアドレスを取得し、取得したアドレスをＥＭＴ２１Ａの“現ＳＭＴポインタ”に設定する。

【００６５】

これによって、性能情報収集部１７は、ＥＭＴ２１Ａから所望のＳＭＴ２２へ直接アクセスすることができる。図３には、ＥＭＴ２１Ａの“現ＳＭＴポインタ”として、ＳＭＴ２２Ｃのアドレスが保持されている場合の様子が示されている。

【００６６】

性能情報収集部１７は、或るＳＭＴ２２から所望のＣＭＴ２３にアクセスする場合には、当該ＳＭＴ２２の“先頭ＣＭＴポインタ”を用いて先頭ＣＭＴに相当するＣＭＴ２３にアクセスする。

【００６７】

続いて、性能情報収集部１７は、先頭ＣＭＴが所望のＣＭＴ２３でない場合には、先頭ＣＭＴ中の“次ＣＭＴポインタ”を用いて次のＣＭＴ２３にアクセスする。このような処理を繰り返すことにより、所望のＣＭＴ２３にアクセスすることができ、且つＳＭＴ２２の“現ＣＭＴポインタ”に所望のＣＭＴ２３のアドレスを設定することができる。

【００６８】

図３には、ＳＭＴ２２Ａ(ＳＭＴ２２Ｂ)の“先頭ＣＭＴポインタ”及び“現Ｃ

ＭＴポインタ”としてＣＭＴ２３Ａ(ＣＭＴ２３Ｂ)のアドレスが保持されている場合の様子が示されている。

【００６９】

制御表１８が上記データ構造を有することにより、各実行体１９によって実行されるサブルーチンの性能情報が実行体１９毎に保管される。また、各ＳＭＴ２２によって、或る実行体によって実行されるサブルーチンの性能情報がサブルーチン毎に保管される。さらに、各ＣＭＴ２３によって、各サブルーチンから呼び出され実行されるサブルーチンの性能情報と、サブルーチン間の呼出関係とが保管される。

【００７０】

なお、図３に示す例では、サブルーチンのネスティングが１回である場合の例を示しているが、ネスティングが２回以上行われる場合には、さらに下位のＣＭＴ２３が作成される。例えば、ＣＭＴ２３Ａで管理される呼出先のサブルーチンから他のサブルーチンが呼び出される場合には、この呼出関係と呼び出された他のサブルーチンの性能情報を保管する下位のＣＭＴ２３が作成される。

【００７１】

さらに、図３に例示した制御表１８は、いわゆる「再帰(recursion)」を考慮しない場合(即ち、サブルーチン自身への再帰は単に分岐とみなし、呼出元に該当するメインルーチン又はサブルーチンに復帰する場合にのみカウントする)のデータ構造例を示す。これに代えて、制御表１８を「再帰」を考慮したデータ構造に拡張することもできる。また、本発明における制御表１８のデータ構造は、任意の構成を採用することができ、検索の効率化を可能とした二次的なハッシュ検索データの構築、制御表間の相互リンク構造の構築などを行なうことが可能である。

【００７２】

なお、制御表１８は、サブルーチンを実行するタスク又はプロセス毎に作成されるようにしても良く、サブルーチンを実行するスレッド毎に作成されるようにしても良い。

【００７３】

〈動作例〉

次に、上述したプログラム性能情報収集装置の動作例を説明する。図２において、最初に、実行環境設定部１５が、ＣＰＵ２に対し、上述した設定を施す(図２(Ａ)参照)。その後、ＣＰＵ２が、ＯＳ８上でアプリケーション９を実行する。これによって、図２(Ｂ)に示すように、アプリケーション９がＯＳ８上で実行され、アプリケーション９中のサブルーチンが実行される場合には、サブルーチンの実行体１９が生成され、実行体１９がサブルーチンを実行する。

【００７４】

アプリケーション９の実行時において、分岐命令が実行されると、ＣＰＵ２が割込を発生させるとともに、制御を命令分析部１６に渡す。これによって、命令分析部１６が割込処理を開始する。図４は、命令分析部１６の処理を示すフローチャートである。

【００７５】

図４において、最初に、命令分析部１６は、実行された分岐命令の分岐元アドレスと分岐先アドレスとをＣＰＵ２から受け取る(ステップＳ１)。次に、命令分析部１６は、分岐元アドレスの命令コードをMM３から読み出し(ステップＳ２)、命令コードの命令フォーマットをデコードすることによって分岐命令の種別を判別する(ステップＳ３)。

【００７６】

続いて、命令分析部１６は、分岐命令を判別した結果、命令種別がサブルーチンの呼出命令(ＣＡＬＬ命令)又は復帰命令(ＲＥＴＵＲＮ命令)であるか否かを判定する(ステップＳ４)。即ち、命令分析部１６は、分岐命令がサブルーチンの実行に関する命令(呼出命令又は復帰命令)であるか否かを判別する。

【００７７】

このとき、分岐命令が呼出命令又は復帰命令である場合には、分岐元アドレス，分岐先アドレス及び命令種別の判別結果を性能情報収集部１７に与える(ステップＳ５)。これに対し、命令分析部１６は、命令種別が呼出命令及び復帰命令以外である場合には、割込処理を終了し、制御をＣＰＵ２に戻す(ステップＳ６)。

【００７８】

処理がステップＳ５へ進んだ場合には、命令分析部１６に続いて、性能情報収集部１７による割込処理が実行される。図５，６，７は、性能情報収集部１７の処理を示すフローチャートである。

【００７９】

図５において、性能情報収集部１７は、命令分析部１６から分岐元アドレス，分岐先アドレス及び命令種別の判別結果を受け取ると、サブルーチンの実行体ＩＤをＯＳ８から取得し、サブルーチンの実行体１９を特定する(ステップＳ１０１)。

【００８０】

次に、性能情報収集部１７は、命令種別の判別結果がサブルーチンの呼出命令か否かを判定する(ステップＳ１０２)。このとき、判別結果が呼出命令である場合には、処理がステップＳ１０３に進み、判別結果が呼出命令でない場合(復帰命令である場合)には、処理が図７のステップＳ１２５へ進む。

【００８１】

ステップＳ１０３では、性能情報収集部１７は、呼出命令がサブルーチン中からの呼出命令，即ち或るサブルーチンから他のサブルーチンを呼び出す呼出命令か否かを判定する。具体的には、性能情報収集部１７は、命令分析部１６から受け取った分岐元アドレスがサブルーチンのアドレスを示すか否かを判定する。

【００８２】

このとき、分岐元アドレスがサブルーチンのアドレスでない場合(メインルーチンのアドレスである場合)には、性能情報収集部１７は、呼出命令がサブルーチン中からの呼出命令ではないものとして、処理をステップＳ１０４に進める。これに対し、分岐元アドレスがサブルーチンのアドレスである場合には、性能情報収集部１７は、呼出命令がサブルーチン中からの呼出命令であるものとして、処理を図６に示すステップＳ１１５に進める。

【００８３】

ステップ１０４に処理が進んだ場合には、性能情報収集部１７は、ステップＳ１０１にて取得した実行体ＩＤを保持したＥＭＴ２１を含む制御表１８を検索す

る。即ち、性能情報収集部１７は、ステップＳ１０１にて特定した実行体１９に対応するＥＭＴ２１があるか否かを判定する。このとき、対応するＥＭＴ２１がない場合には、性能情報収集部１７は、処理をステップＳ１０５に進め、対応するＥＭＴ２１がある場合には、処理をステップＳ１０７に進める。

【００８４】

ステップＳ１０５では、性能情報収集部１７は、実行体１９に対応する制御表１８がないものとして、実行体１９に対応するＥＭＴ２１を新規に作成する。これにより、実行体１９に対応した新規の制御表１８が作成される。続いて、性能情報収集部１７は、作成したＥＭＴ２１にステップＳ１０１にて取得した実行体ＩＤを設定(登録)する(ステップＳ１０６)。その後、処理がステップＳ１０７に進む。

【００８５】

ステップＳ１０７では、性能情報収集部１７は、呼出命令によって呼び出されたサブルーチンに対応するＳＭＴ２２があるか否かを判定する。即ち、性能情報収集部１７は、命令分析部１６から受け取った分岐先アドレスを“サブルーチンアドレス”として保持したＳＭＴ２２を検索する。このとき、該当するＳＭＴ２２がある場合には、処理がステップＳ１１３に進み、該当するＳＭＴ２２がない場合には、処理がステップＳ１０８に進む。

【００８６】

ステップＳ１０８では、性能情報収集部１７は、サブルーチンに対応する新規なＳＭＴ２２を作成する。続いて、性能情報収集部１７は、分岐先アドレスを“サブルーチンアドレス”として、作成したＳＭＴ２２に設定(登録)する(ステップＳ１０９)。

【００８７】

次に、性能情報収集部１７は、新規に作成したＳＭＴ２２が先頭ＳＭＴに相当するか否かを判定する(ステップＳ１１０)。このとき、ＳＭＴ２２が先頭ＳＭＴに相当する場合には、処理がステップＳ１１１に進み、そうでない場合には、処理がステップＳ１１２に進む。

【００８８】

ステップＳ１１１に処理が進んだ場合には、性能情報収集部１７は、新規に作成したＳＭＴ２２のアドレスを、“先頭ＳＭＴポインタ”として、対応するＥＭＴ２１に設定する。その後、処理がステップＳ１１３に進む。

【００８９】

ステップＳ１１２に処理が進んだ場合には、性能情報収集部１７は、既存のＳＭＴ２２の何れかと新規に作成したＳＭＴ２２とのリンクを設定し、各ＳＭＴ２２の“ＳＭＴ双方向リンクポインタ”に適宜のアドレスを設定(登録)する。その後、処理がステップＳ１１３に進む。

【００９０】

ステップＳ１１３では、性能情報収集部１７は、呼び出されたサブルーチンのアドレスが登録されたＳＭＴ２２のアドレスを、“現ＳＭＴポインタ”として、対応するＥＭＴ２１に設定(登録)する。

【００９１】

続いて、性能情報収集部１７は、例えば、現在の時刻を、“最後の呼出時刻”として、該当するＳＭＴ２２に格納する(ステップＳ１１４)。なお、時刻は、コンピュータ１に搭載された図示せぬ内蔵時計から受け取ることができる。その後、性能情報収集部１７は、割込処理を終了し、制御をＣＰＵ２に戻す。その後、実行体１９によってサブルーチンが実行される。

【００９２】

ところで、処理が図６のステップＳ１１５に進んだ場合には、性能情報収集部１７は、呼出先のサブルーチンに対応するＣＭＴ２３があるか否かを判定する。即ち、性能情報収集部１７は、命令分析部１６から受け取った分岐先アドレスを“分岐先アドレス”として保持したＣＭＴ２３を検索する。

【００９３】

このとき、性能情報収集部１７は、該当するＣＭＴ２３が見つかった場合には、処理をステップＳ１１８に進め、該当するＣＭＴ２３が見つからない場合には、呼出先のサブルーチンに対応するＣＭＴ２３がないものとして、処理をステップＳ１１６に進める。

【００９４】

ステップＳ１１６では、性能情報収集部１７は、呼出先のサブルーチンに対応するＣＭＴ２３を新規に作成する。続いて、性能情報収集部１７は、命令分析部１６から受け取った分岐元アドレスを、“呼出元のサブルーチンアドレス”として、作成したＣＭＴ２３に設定(登録)し、且つ命令分析部１６から受け取った分岐先アドレスを、“呼出先のサブルーチンアドレス”として、作成したＣＭＴ２３に設定(登録)する(ステップＳ１１７)。これによって、サブルーチンのネスティングが行われた場合におけるサブルーチン間の呼出関係がＣＭＴ２３に格納される。その後、処理がステップＳ１１８に進む。

【００９５】

ステップＳ１１８では、性能情報収集部１７は、作成したＣＭＴ２３が“先頭ＣＭＴ”に該当するか否かを判定する。このとき、当該ＣＭＴ２３が“先頭ＣＭＴ”に該当する場合には、性能情報収集部１７は、当該ＣＭＴ２３のアドレスを、“先頭ＣＭＴポインタ”として、呼出元のサブルーチンに対応するＳＭＴ２２に設定(登録)し(ステップＳ１１９)、その後、処理をステップＳ１２１へ進める。

【００９６】

これに対し、当該ＣＭＴ２３が“先頭ＣＭＴ”に該当しない場合には、性能情報収集部１７は、当該ＣＭＴ２３のアドレスを、“次のＣＭＴポインタ”として、当該ＣＭＴ２３の１つ前に配置された他のＣＭＴ２３に設定(登録)し(ステップＳ１２０)、その後、処理をステップＳ１２１へ進める。

【００９７】

ステップＳ１２１では、性能情報収集部１７は、呼出先のサブルーチンに対応するＳＭＴ２２があるか否かを判定する。即ち、性能情報収集部１７は、当該ＣＭＴ２３に“分岐先アドレス(呼出先のサブルーチンアドレス)”として設定したアドレスが“サブルーチンアドレス”として登録されたＳＭＴ２２を、当該制御表１８から検索する。

【００９８】

このとき、該当するＳＭＴ２２がない場合には、性能情報収集部１７は、処理をステップＳ１２３に進める。これに対し、該当するＳＭＴ２２が見つかった場

合には、性能情報収集部１７は、見つかったＳＭＴ２２のアドレスを、“呼出先ＳＭＴポインタ”として、当該ＣＭＴ２３に設定(登録)し(ステップＳ１１９)、その後、処理をステップＳ１２３に進める。

【００９９】

ステップＳ１２３では、性能情報収集部１７は、当該ＣＭＴ２３のアドレスを、“現ＣＭＴポインタ”として、呼出元のサブルーチンに対応するＳＭＴ２２に設定(登録)する。

【０１００】

続いて、性能情報収集部１７は、例えば、現時刻を取得し、取得した現時刻を、“最後の呼出時刻”として、当該ＣＭＴ２３に格納する(ステップＳ１２４)。その後、性能情報収集部１７は、割込処理を終了し、制御をＣＰＵ２に戻す。その後、ネスティングによって呼び出されたサブルーチンが実行体１９によって実行される。

【０１０１】

ところで、分岐命令の判別結果が復帰命令であり、処理が図７に示したステップＳ１２５に進んだ場合には、性能情報収集部１７は、復帰命令が他のサブルーチンへの復帰命令か否かを判定する。具体的には、性能情報収集部１７は、命令分析部１６から受け取った分岐先アドレスがサブルーチンのアドレスを示すか否かを判定する。

【０１０２】

このとき、分岐先アドレスがサブルーチンのアドレスでない場合(メインルーチンのアドレスである場合)には、性能情報収集部１７は、復帰命令が他のサブルーチンへの復帰命令ではないものとして、処理をステップＳ１２６に進める。これに対し、分岐先アドレスがサブルーチンのアドレスである場合には、性能情報収集部１７は、復帰命令が他のサブルーチンへの復帰命令であるものとして、処理をステップＳ１３０に進める。

【０１０３】

ステップＳ１２６では、性能情報収集部１７は、該当するＳＭＴ２２の“呼出回数”を１インクリメントする。続いて、性能情報収集部１７は、現在の時刻を

取得し(ステップＳ１２７)、現時刻と、当該ＳＭＴ２２に格納された“最後の呼出時刻”とから、サブルーチンの実行時間を算出する(ステップＳ１２８)。

【０１０４】

続いて、性能情報収集部１７は、当該ＳＭＴ２２の“累計実行時間”の値に、ステップＳ１２８にて算出した実行時間を加算することによって、“累計実行時間”を更新する(ステップＳ１２９)。その後、性能情報収集部１７は、割込処理を終了し、制御をＣＰＵ２に戻す。

【０１０５】

一方、ステップＳ１３０に処理が進んだ場合には、性能情報収集部１７は、該当するＣＭＴ２３の“呼出回数”を１インクリメントする。続いて、性能情報収集部１７は、現在の時刻を取得し(ステップＳ１３１)、現時刻と、当該ＣＭＴ２３に格納された“最後の呼出時刻”とから、サブルーチンの実行時間を算出する(ステップＳ１３２)。

【０１０６】

続いて、性能情報収集部１７は、当該ＣＭＴ２３の“累計実行時間”の値に、ステップＳ１３２にて算出した実行時間を加算することによって、“累計実行時間”を更新する(ステップＳ１３３)。その後、性能情報収集部１７は、割込処理を終了し、制御をＣＰＵ２に戻す。

【０１０７】

〈実施形態の作用〉

上述した実施形態によるプログラム性能情報収集装置によると、実行環境設定部１５によるＣＰＵ２の設定により、アプリケーション９の実行時に分岐命令が実行されると、割込が発生し、制御がＣＰＵ２から命令分析部１６へ渡される。

【０１０８】

その後、命令分析部１６及び性能情報収集部１７によって、サブルーチンの実行体１９毎に制御表１８が作成され、各制御表１８において、実行体１９によって実行される各サブルーチンの性能情報(プロフィール)が保管される。

【０１０９】

即ち、制御表１８における各ＳＭＴ２２において、メインルーチンから呼び出

されたサブルーチンの呼出回数，累計実行時間及び最後の呼出時刻が性能情報として保持される。また、制御表１８における各ＣＭＴ２３において、或るサブルーチンから呼び出された他のサブルーチンの呼出回数，累計実行時間及び最後の呼出時刻が性能情報として保持される。また、各ＣＭＴ２３には、呼出元のサブルーチンと呼出先のサブルーチンとの夫々のアドレスが格納されることにより、サブルーチン間の呼出関係を示す情報(呼出関係情報)が保持される。

【０１１０】

上記した命令分析部１６及び性能情報収集部１７の処理(分析プログラム１０の実行による処理)は、ＣＰＵ２の機能を用いた割込により行うため、アプリケーション９に対して何ら改変を加える必要がない。従って、制御コードをプログラムに埋め込むことによるオーバヘッドの増加や、プログラムの改変を起因とするプログラムの不動作といった従来の問題は生じない。

【０１１１】

アプリケーション９の実行が終了すると、ＭＭ３又は二次記憶装置４上には、アプリケーション９に含まれた複数のサブルーチンについての性能情報及び呼出関係情報を保持した複数の制御表１８が作成される。コンピュータ１のオペレータは、入力装置１１の操作によって、各制御表１８の保持内容をディスプレイ７に表示したり、図示せぬプリンタによって紙等のシートに印刷したりすることができる。そして、ディスプレイ７に表示されたり、シートに印刷されたりした制御表１８の保持内容は、アプリケーション９の改良や、新規のアプリケーションプログラムの作成の資料として使用される。

【０１１２】

なお、本実施形態では、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報をＳＭＴ２２に格納し、他のサブルーチンから呼び出された場合の性能情報をＣＭＴ２３に格納した。これらのＳＭＴ２２及びＣＭＴ２３中の呼出回数同士，累計実行時間同士を夫々加算することにより、或るサブルーチンのアプリケーション９の実行時における累計実行時間を得ることができる。

【０１１３】

これに対し、ＳＭＴ２２には、或るサブルーチンの呼出回数及び累計実行時間呼出回数を格納し、ＣＭＴ２３にＳＭＴ２２に格納された呼出回数及び累計実行時間のうち、他のサブルーチンから呼び出された場合の呼出回数及び累計実行時間を格納するようにしても良い。この場合には、ＳＭＴ２２に格納された呼出回数及び累計実行時間からＣＭＴ２３に格納された呼出回数及び累計実行時間を夫々減算することにより、メインルーチンから呼び出された場合の呼出回数及び累計実行時間を得ることができる。

【０１１４】

〔第２実施形態〕

次に、本発明の第２実施形態によるプログラム性能情報収集装置について説明する。第２実施形態は、第１実施形態の構成に加え、サブルーチンの性能情報として、オーバヘッドを制御表１８に格納するようにしたものである。

【０１１５】

即ち、第２実施形態では、性能情報収集部１７は、アプリケーション９の実行時において、分岐命令の実行を契機とする割込処理の発生回数を、各サブルーチンについて、その呼出元アドレス(分岐元アドレス)に従い、各ＳＭＴ２３及び各ＣＭＴ２３に夫々格納する。

【０１１６】

即ち、性能情報収集部は、各サブルーチンに対する呼出命令，復帰命令及びジャンプ命令の発生回数の合計値を、そのサブルーチンがメインルーチンから呼び出された場合と、他のサブルーチンとから呼び出された場合とに分けて、該当するＳＭＴ２２及びＣＭＴ２３に格納する。また、性能情報収集部１７は、平均オーバヘッド時間を算出する。図８に第２実施形態における制御表１８Ａの構成例を示す。

【０１１７】

その後、アプリケーション９の実行中，或いは実行終了後に、各ＳＭＴ２２及び各ＣＭＴ２３に格納された割込処理の発生回数に、算出した平均オーバヘッド時間を乗じ、この結果を各サブルーチンの実行時におけるオーバヘッド時間として各ＳＭＴ２２及び各ＣＭＴ２３に格納する。

【０１１８】

第２実施形態によれば、サブルーチンの性能情報として、さらにオーバヘッドが各制御表１８に格納されるので、第１実施形態よりも詳細なサブルーチンの性能情報(プロフィール)を得ることができ、性能情報の精度を向上させることができる。

【０１１９】

〔付記〕

本発明は、以下のように特定することができる。

（付記１）プログラムに含まれたサブルーチンの性能情報を収集する装置であって、前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する、プログラム性能情報収集装置。

（付記２）サブルーチンの実行体毎に前記性能情報を収集する、付記１記載のプログラム性能情報収集装置。

（付記３）特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する、付記２記載のプログラム性能情報収集装置。

（付記４）或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する、付記３記載のプログラム性能情報収集装置。

（付記５）前記他のサブルーチンから呼び出された場合の性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて保持する、付記４記載のプログラム性能情報収集装置。

（付記６）前記性能情報は、サブルーチンの累計実行時間，呼出回数，最後に呼び出された時刻，オーバヘッドのうちの少なくとも１つである、付記１記載のプログラム性能情報収集装置。

（付記７）前記サブルーチンの性能情報を保持する性能情報記憶部と、前記割込が発生した場合に、実行された分岐命令が前記サブルーチンの実行に関する命令か否かを判別する命令分析部と、前記命令分析部によって前記分岐命令が前記サブルーチンの実行に関する命令と判別した場合に、前記サブルーチンの性能情報

を取得して前記性能情報記憶部に格納する性能情報収集部と、を備えた付記１記載のプログラム性能情報収集装置。

（付記８）前記性能情報記憶部は、前記サブルーチンの実行体毎に設けられ、前記性能情報収集部は、前記サブルーチンの実行体を特定し、特定した実行体に対応する性能情報記憶部に前記性能情報を格納する、付記７記載のプログラム性能情報収集装置。

（付記９）前記性能情報収集部は、特定の実行体によって複数のサブルーチンが実行される場合に、各サブルーチンの性能情報を、前記特定の実行体に対応する性能情報記憶部に個別に格納する、付記８記載のプログラム性能情報収集装置。

（付記１０）前記性能情報収集部は、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを、前記性能情報記憶部に個別に格納する、付記９記載のプログラム性能情報収集装置。

（付記１１）前記性能情報収集部は、他のサブルーチンから呼び出された場合のサブルーチンの性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて前記性能情報記憶部に格納する、付記１０記載のプログラム性能情報収集装置。

（付記１２）前記性能情報を記憶する性能情報記憶部と、前記割込が発生した場合に、前記割込の発生源から分岐元アドレス及び分岐先アドレスとを受け取り、分岐元アドレスから命令コードを取得しデコードすることによって分岐命令の種別を判別する命令分析部と、判別された種別がサブルーチンの呼出命令又は復帰命令である場合に、分岐元アドレス，分岐先アドレス及び判別結果を前記命令分析部から受け取り、受け取った判別結果が呼出命令である場合には、受け取った分岐先アドレスを呼出命令に対応するサブルーチンアドレスとして前記性能情報記憶部に記憶するとともに、呼出命令に対応するサブルーチンの呼出時刻を前記分岐先アドレスに関連づけて前記性能情報記憶部に格納し、受け取った判別結果が復帰命令である場合には、復帰命令に対応するサブルーチンの復帰時刻を取得し、取得した復帰時刻と前記呼出時刻とから前記サブルーチンの実行時間を算出し、実行時間の累計値を前記性能情報として前記分岐先アドレスと関連づけて前

記性能情報記憶部に格納する、付記１記載のプログラム性能情報収集装置。

（付記１３）前記性能情報収集部は、サブルーチンの呼出回数を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納する、付記１２記載のプログラム性能情報収集装置。

（付記１４）前記性能情報収集部は、前記性能情報として、サブルーチンのオーバヘッドを取得し前記性能情報記憶部に格納する、付記１２記載のプログラム性能情報収集装置。

（付記１５）前記性能情報収集部は、受け取った判別結果がサブルーチンの呼出命令である場合には、この呼出命令に対応するサブルーチンの実行体の識別情報を取得し、前記分岐先アドレスと関連づけて前記性能情報記憶部に格納する、付記１３記載のプログラム性能情報収集装置。

（付記１６）前記性能情報収集部は、受け取った判別結果がサブルーチンの呼出命令であり、且つ受け取った分岐元アドレス及び分岐先アドレスがサブルーチンのアドレスを示す場合には、これらの分岐元アドレス及び分岐先アドレスを呼出元のサブルーチンと呼出先のサブルーチンとを示す呼出関係情報として前記性能情報記憶部に格納するとともに、呼出元のサブルーチンにおける呼出先のサブルーチンの累計実行時間と呼出回数との少なくとも一方を、前記性能情報として、前記呼出関係情報と関連づけて前記性能情報記憶部に格納する、付記１３記載のプログラム性能情報収集装置。

（付記１７）前記プログラムの実行中に分岐命令が実行されたときに割込を発生させるように割込発生源の実行環境を設定する実行環境設定部をさらに備えた、付記７記載のプログラム性能情報収集装置。

（付記１８）分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって、コンピュータに、前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可

能な記録媒体。

（付記１９）前記サブルーチンの実行体を特定するステップと、実行体毎に設けられた複数の性能情報記憶部のうち、特定された実行体に対応する性能情報記憶部に前記性能情報を格納するステップと、をさらに実行させる前記プログラムを記録した付記１８記載のコンピュータ読取可能な記録媒体。

（付記２０）特定の実行体によって複数のサブルーチンが実行される場合に、各サブルーチンの性能情報を、前記特定の実行体に対応する性能情報記憶部に個別に格納するステップ、をさらに実行させる前記プログラムを記録した付記１９記載のコンピュータ読取可能な記録媒体。

（付記２１）前記各サブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを、前記性能情報記憶部に個別に格納するステップ、をさらに実行させる前記プログラムを記録した付記２０記載のコンピュータ読取可能な記録媒体。

（付記２２）他のサブルーチンから呼び出された場合のサブルーチンの性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記２１記載のコンピュータ読取可能な記録媒体。

（付記２３）割込が発生した場合に前記割込の発生源から分岐元アドレス及び分岐先アドレスとを受け取るステップと、分岐元アドレスから命令コードを取得しデコードすることによって分岐命令の種別を判別するステップと、判別された種別がサブルーチンの呼出命令である場合に、受け取った分岐先アドレスを呼出命令に対応するサブルーチンアドレスとして前記性能情報記憶部に記憶するとともに、呼出命令に対応するサブルーチンの呼出時刻を前記分岐先アドレスに関連づけて前記性能情報記憶部に格納するステップと、判別された種別が復帰命令である場合に、復帰命令に対応するサブルーチンの復帰時刻を取得し、取得した復帰時刻と前記呼出時刻とから前記サブルーチンの実行時間を算出し、実行時間の累計値を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップと、をさらに実行させる前記プログラムを記録した付記１８記載のコンピュータ読取可能な記録媒体。

（付記２４）サブルーチンの呼出回数を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記２３記載のコンピュータ読取可能な記録媒体。

（付記２５）サブルーチンのオーバヘッドを前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記２３又は２４記載のコンピュータ読取可能な記録媒体。

（付記２６）判別された種別がサブルーチンの呼出命令である場合に、呼出命令に対応するサブルーチンの実行体の識別情報を取得し、前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記２３記載のコンピュータ読取可能な記録媒体。

（付記２７）判別された種別がサブルーチンの呼出命令であり、且つ分岐元アドレス及び分岐先アドレスがサブルーチンのアドレスを示す場合に、これらの分岐元アドレス及び分岐先アドレスを呼出元のサブルーチンと呼出先のサブルーチンとを示す呼出関係情報として前記性能情報記憶部に格納するステップと、前記呼出元のサブルーチンにおける前記呼出先のサブルーチンの累計実行時間と呼出回数との少なくとも一方を、前記呼出関係情報と関連づけて前記性能情報記憶部に格納するステップと、をさらに実行させる付記２３記載のコンピュータ読取可能な記録媒体。

（付記２８）前記プログラムの実行中に分岐命令が実行されたときに割込を発生させるように割込発生源の実行環境を設定するステップ、をさらに実行させる前記プログラムを記録した付記１８記載のコンピュータ読取可能な記録媒体。

【０１２０】

【発明の効果】

本発明によれば、プログラムに改変を加えることなくサブルーチンの性能情報を収集することができる。また、詳細なサブルーチンの性能情報を収集することができる。

【図面の簡単な説明】

【図１】プログラム性能情報収集装置として機能するコンピュータのハードウェ

ア構成図

【図２】プログラム性能情報収集装置の機能ブロック図

【図３】制御表の構成例を示す図

【図４】命令分析部の処理を示すフローチャート

【図５】性能情報収集部の処理を示すフローチャート

【図６】性能情報収集部の処理を示すフローチャート

【図７】性能情報収集部の処理を示すフローチャート

【図８】第２実施形態における制御表の構成例を示す図

【符号の説明】

１　コンピュータ

２　ＣＰＵ

３　メインメモリ

４　二次記憶装置

５　キーボード

６　ポインティングデバイス

７　ディスプレイ

８　オペレーティングシステム

９　アプリケーションプログラム

１０　性能分析用プログラム

１１　入力装置

１５　実行環境設定部

１６　命令分析部

１７　性能情報収集部

１８　制御表

２１　実行体管理表(ＥＭＴ)

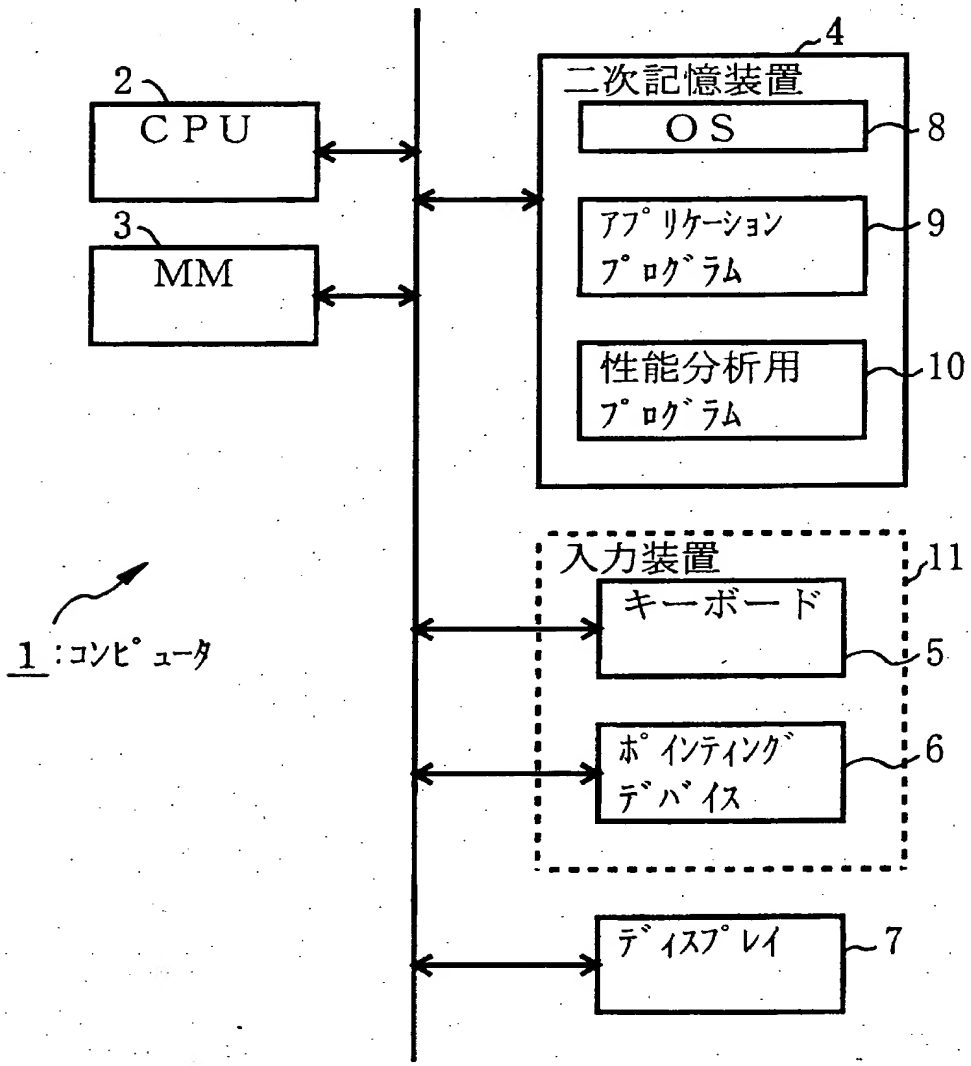
２２　サブルーチン管理表(ＳＭＴ)

２３　呼出管理表(ＣＭＴ)

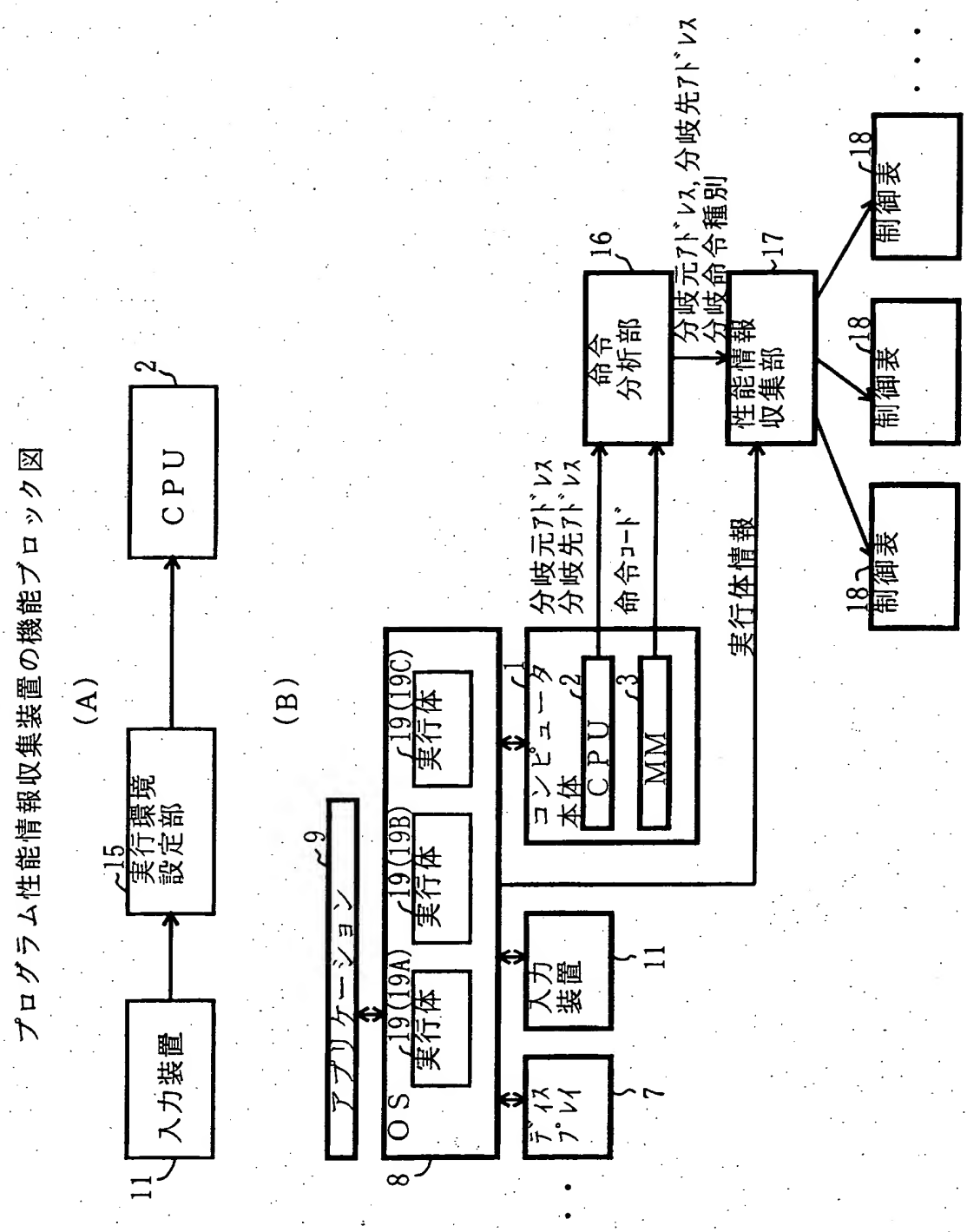
【書類名】　　図面

【図１】

プログラム性能情報収集装置として機能する
コンピュータのハードウェア構成図

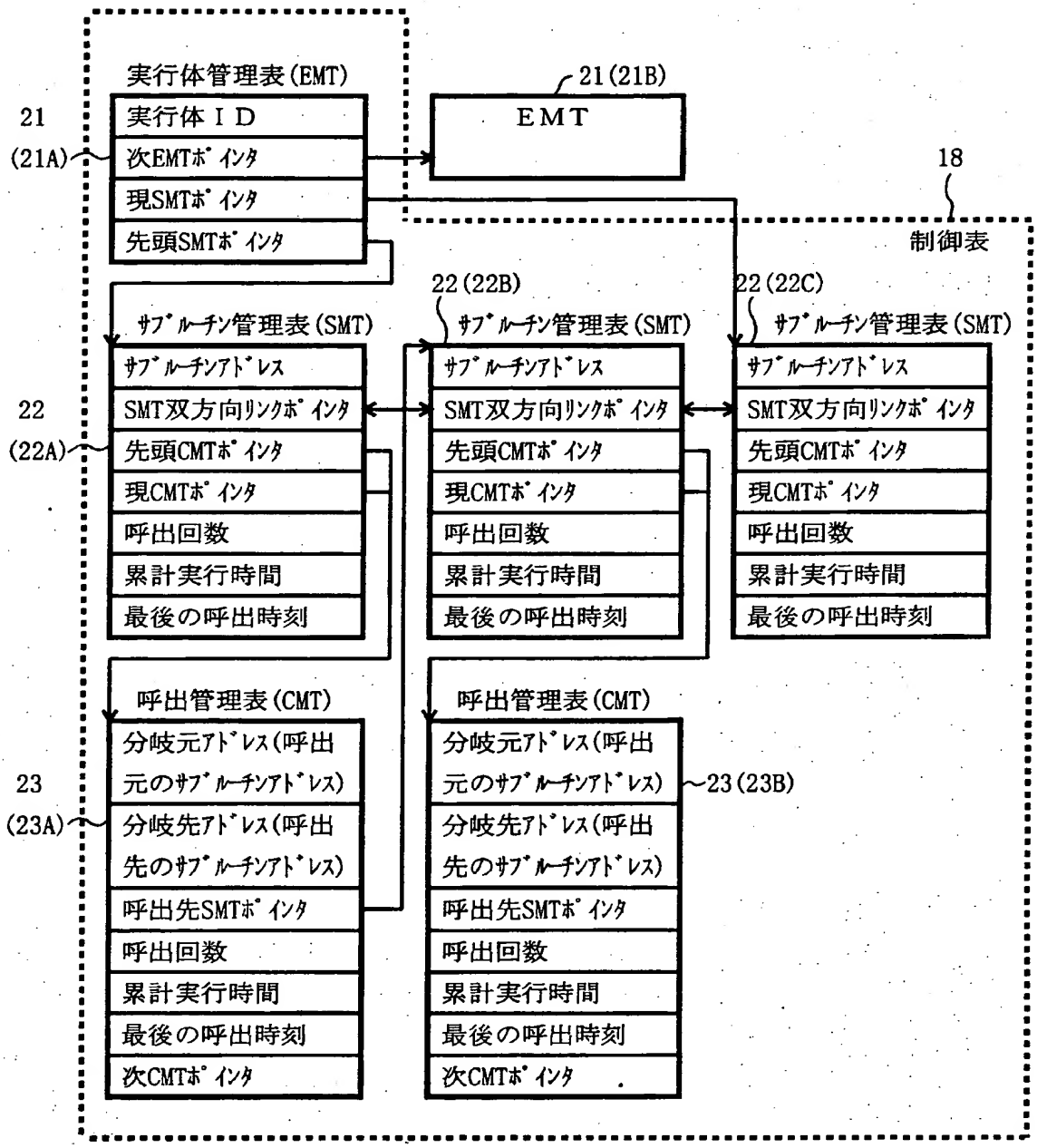


【図２】


プログラム性能情報収集装置の機能ブロック図
(A)
(B)
入力装置
11
実行環境設定部
15
CPU
2
アプリケーション
9
OS
8
19(19A)
実行体
19(19B)
実行体
19(19C)
実行体
ﾃﾞｨｽﾌﾟﾚｲ
7
入力装置
11
コンピュータ本体
1
CPU
2
MM
3
分岐元ｱﾄﾞﾚｽ
分岐先ｱﾄﾞﾚｽ
命令ｺｰﾄﾞ
命令分析部
16
分岐元ｱﾄﾞﾚｽ,分岐先ｱﾄﾞﾚｽ
分岐命令種別
実行体情報
性能情報収集部
17
制御表
18
制御表
18
制御表
18

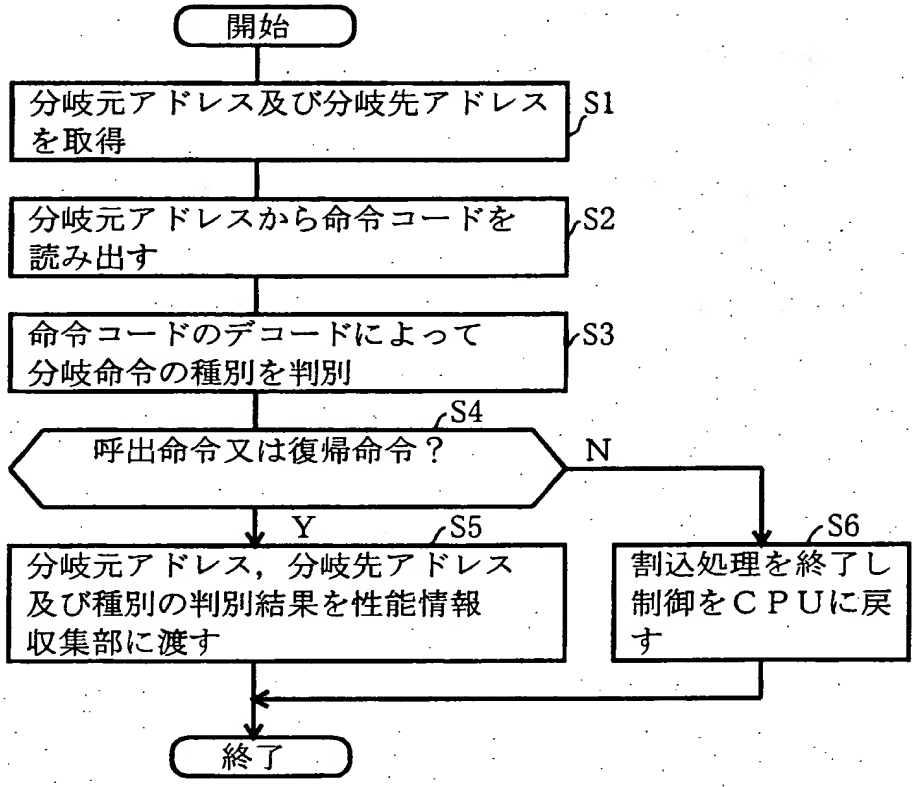
【図３】

制御表の構成例を示す図



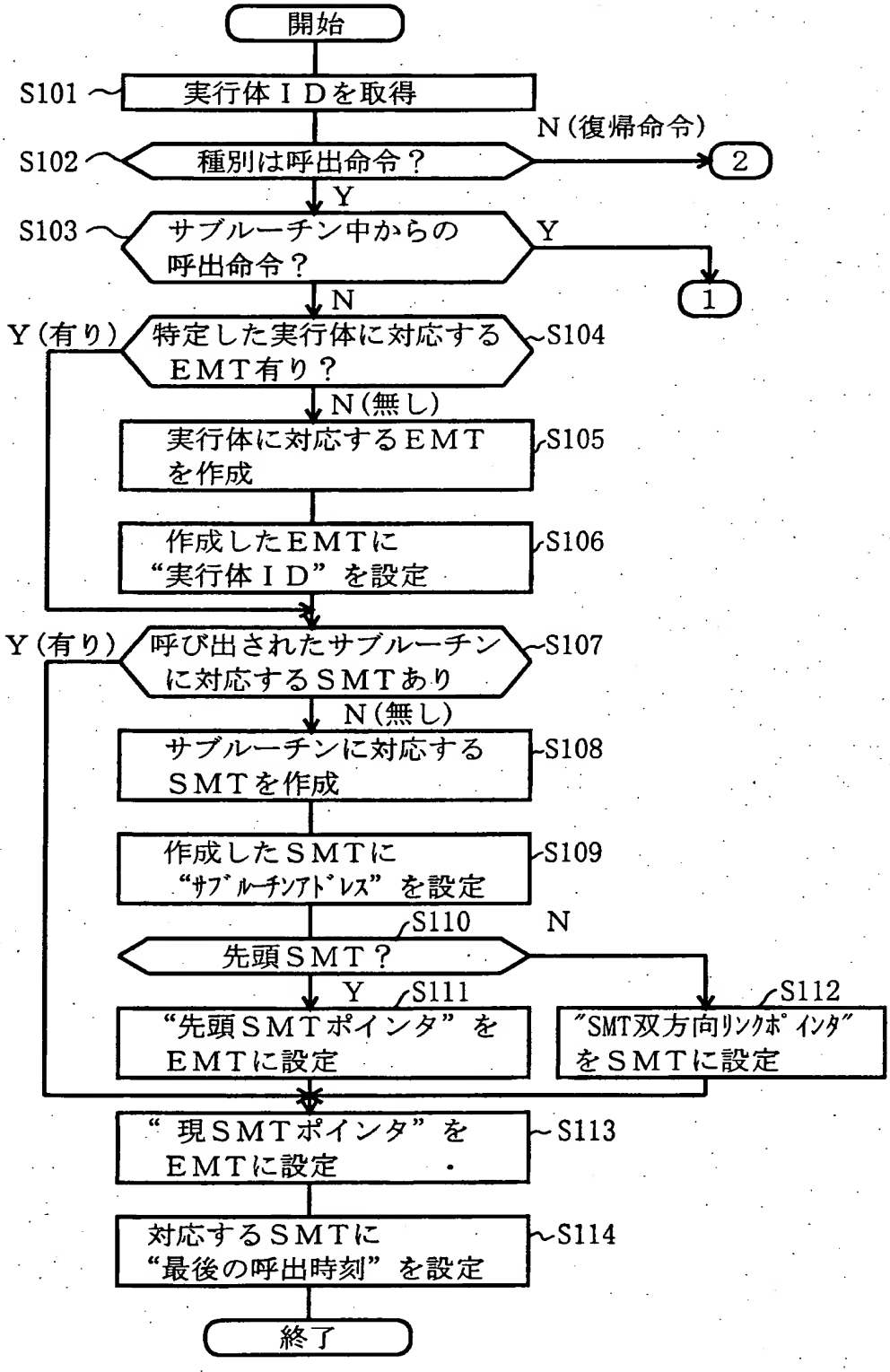
【図４】

命令分析部の処理を示すフローチャート



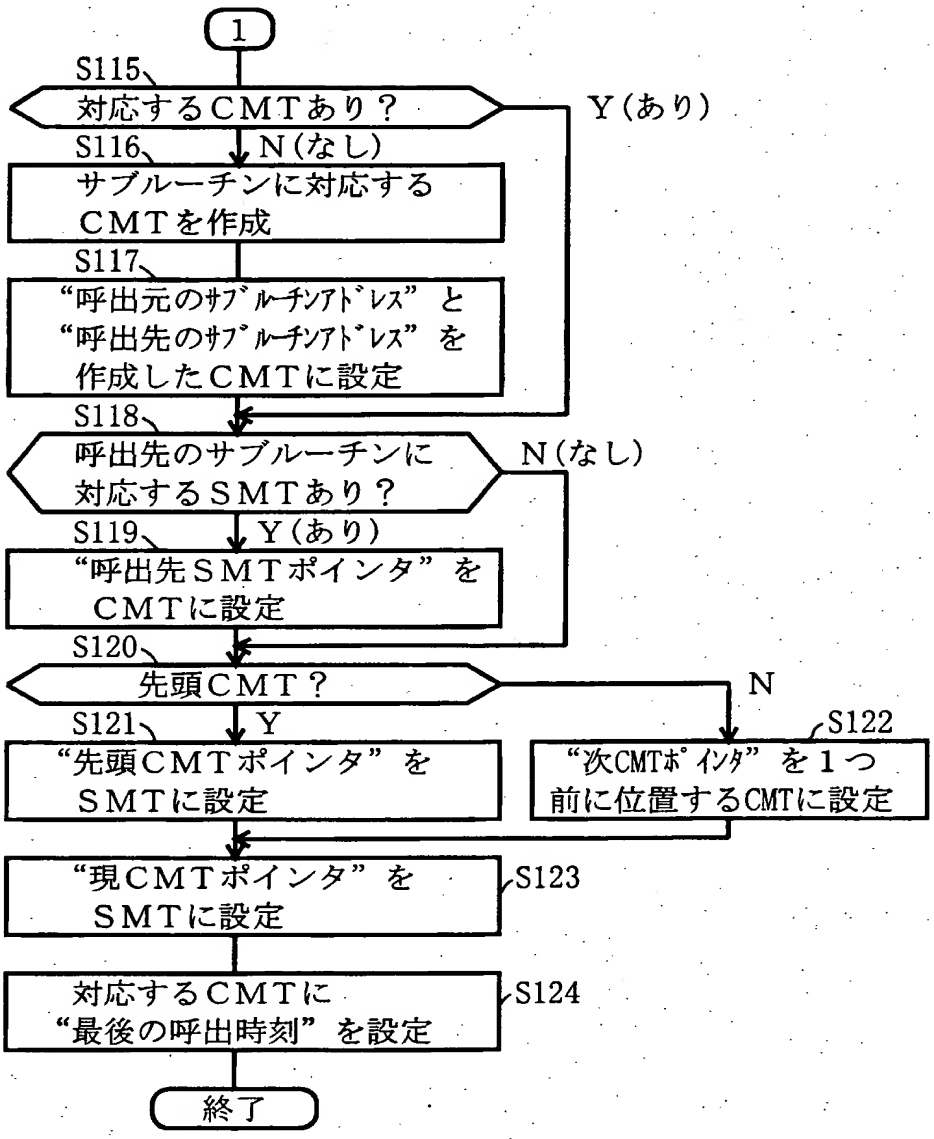
【図５】

性能情報収集部の処理を示すフローチャート



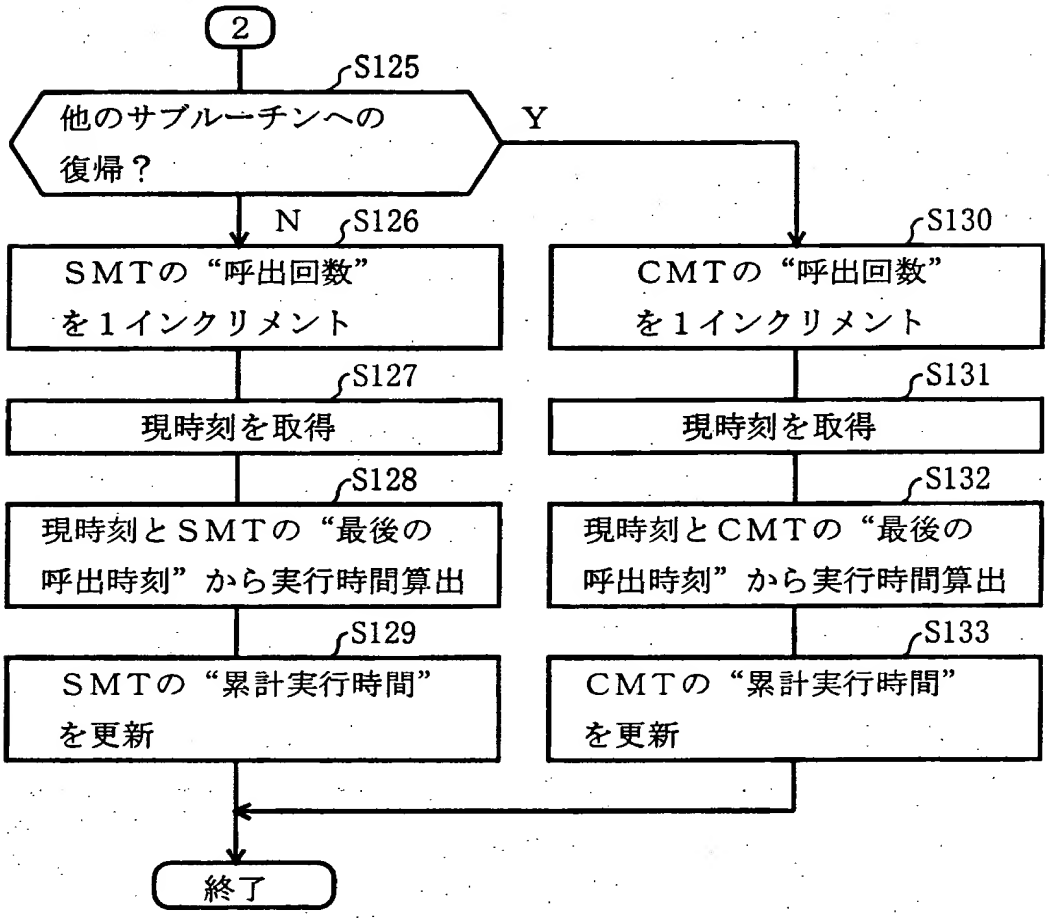
【図６】

性能情報収集部の処理を示すフローチャート



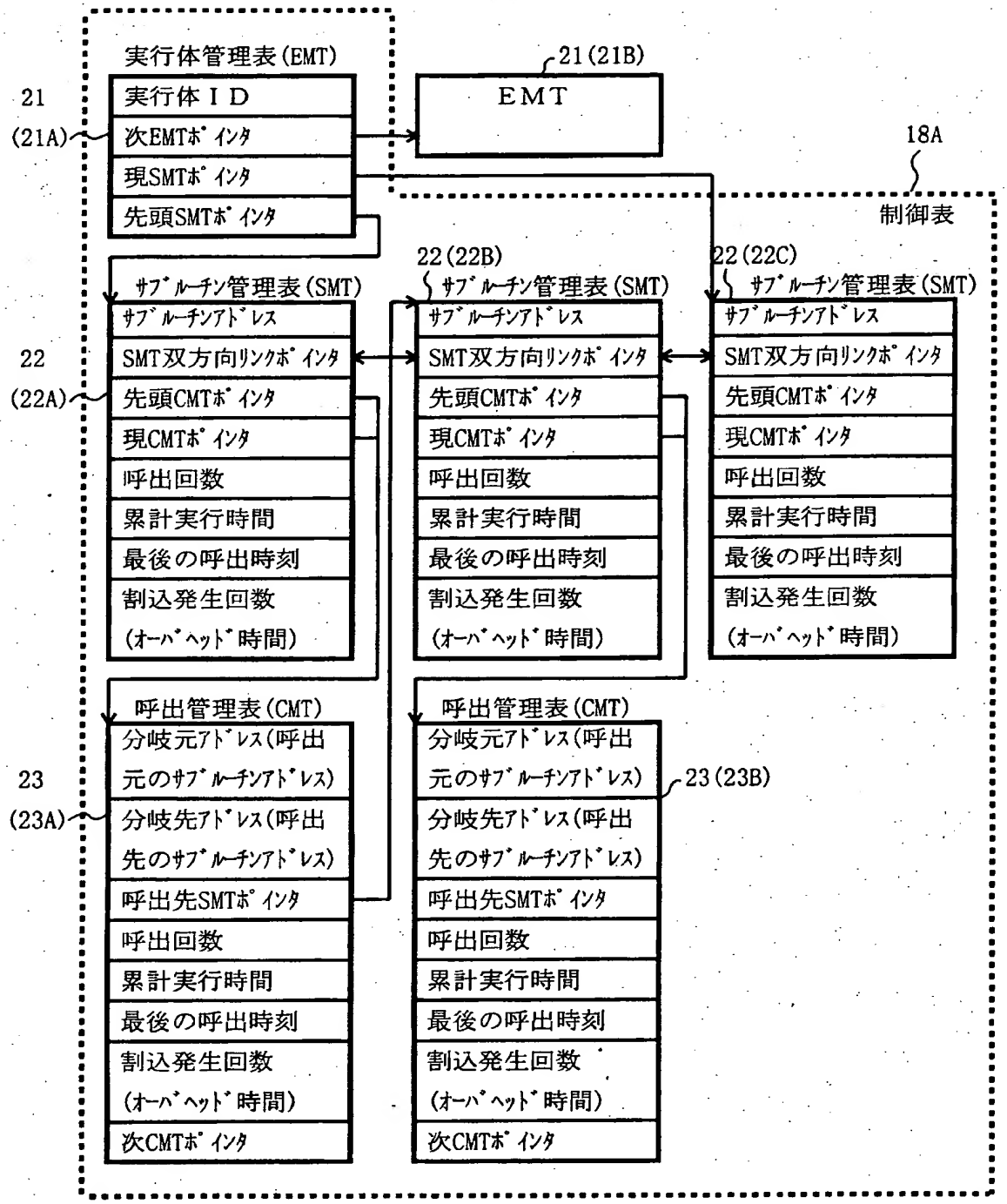
【図７】

性能情報収集部の処理を示すフローチャート



【図８】

制御表の構成例を示す図



【書類名】要約書

【要約】

【課題】プログラムに改変を加えることなくサブルーチンの性能情報を収集することができるプログラム性能情報収集装置を提供する。

【解決手段】プログラム性能情報収集装置１によると、プログラムの実行中に分岐命令が実行されると、ＣＰＵ２による割込が発生する。この割込において、命令分析部１６が、分岐命令がサブルーチンの実行に関する命令か否かを判別し、分岐命令が前記サブルーチンの実行に関する命令である場合には、性能情報収集部１７が、サブルーチンの性能情報を取得して性能情報記憶部１８に格納する。

【選択図】図２

出証特２０００－３０９３３８８

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社